

ACR Enumeration

Name: Terry Cole

Title: Fellow

Company: AMD

r3



San Jose January 23-24, 2001



Taipei February 14-15, 2001

ACR Enumeration

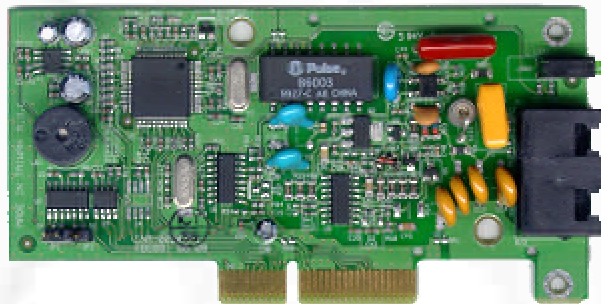
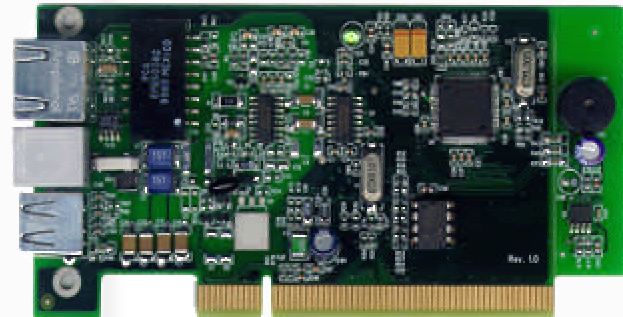
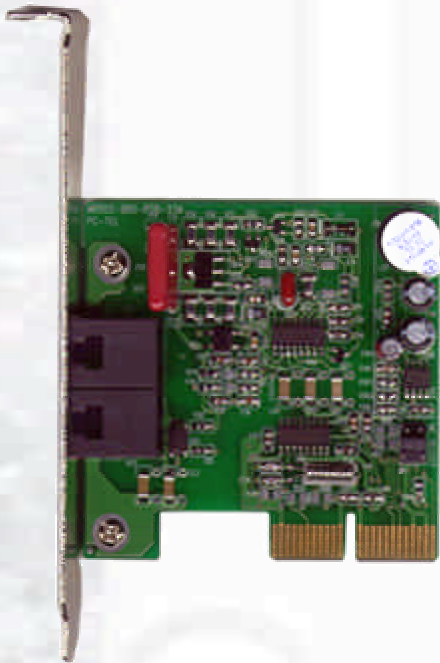
- **Introduction**
- **Enumeration**
 - Architectural role
 - Value
 - Procedure
 - Examples
 - Going further
- **Call to action**

Introduction to Risers

- **Risers have been traditionally used to deliver functions that require connectors outside the PC**
 - **More integrated than PCI cards, external USB peripherals, etc. (And typically less cost)**
 - **Less integrated and more flexible than mother-board down solutions (and typically more cost)**
 - **“Motherboard down” means soldered to planar**
- **Often used to deliver soft audio & modem codecs from multiple IP sources**
- **Can also be designed to deliver more complex functions like entire networking, DSL, cable solutions**

Example risers

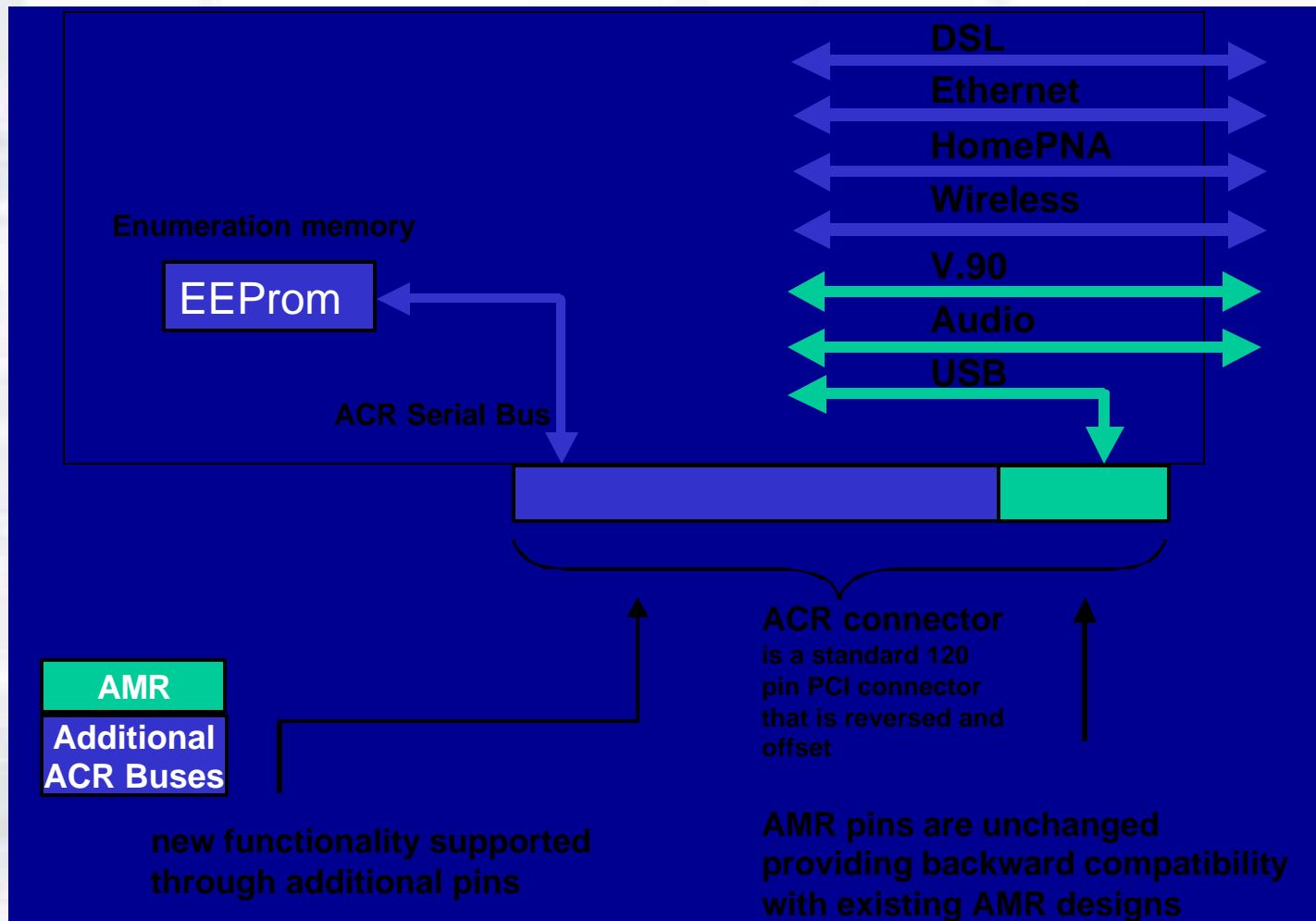
- AMR, MDC, CNR, ACR
- And don't forget serial/parallel ports risers & usb ports risers



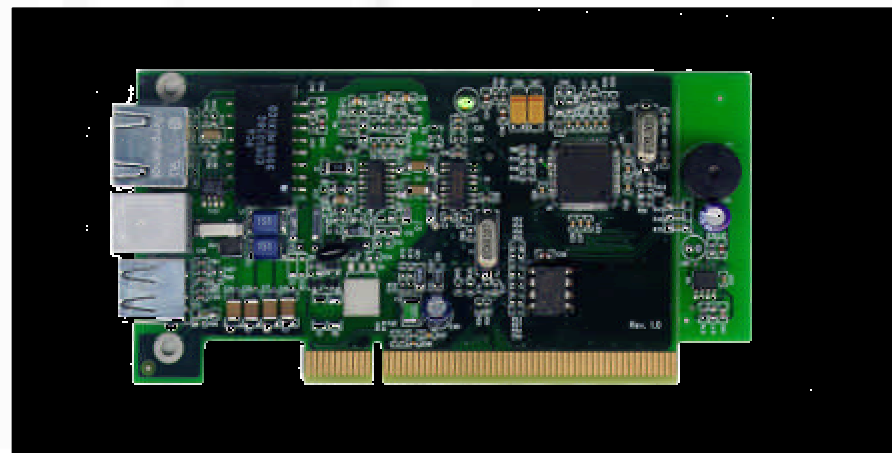
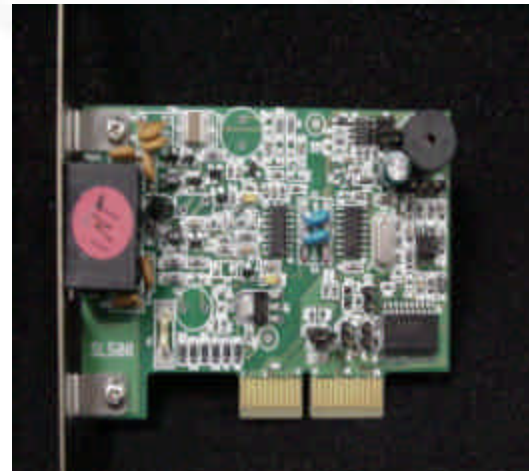
Introduction to ACR

- **ACR = Advanced Communication Riser**
 - A 2nd generation riser
 - AMR is the 1st generation riser
- **Defined by an industry special interest group open to all in the industry**
- **Goals of desktop technology of ACR SIG**
 - Foster diverse and scalable communications, networking and audio solutions using a flexible combination of hardware and software cores
 - Reuse existing industry technology, components
 - Compatibility with existing OS; no significant changes
 - Fit into existing compliance test strategies

ACR Riser



ACR Risers



ACR Enumeration

- Introduction
- **Enumeration**
 - **Architectural role**
 - Value
 - Procedure
 - Examples
 - Functional breakdown
- Call to action

Architecture Role

- To comply with the ACR spec, ACR systems should be designed to provide an enumeration mechanism to identify all ACR components to the BIOS, OS, and software device drivers.
- This information is necessary to ensure that the device operating parameters are correctly set by the BIOS, the correct set of drivers is loaded by the OS, and additional vendor-defined information is available to the device driver.
- Architectural functions
 - Inventory
 - “Marriage”
 - Configuration

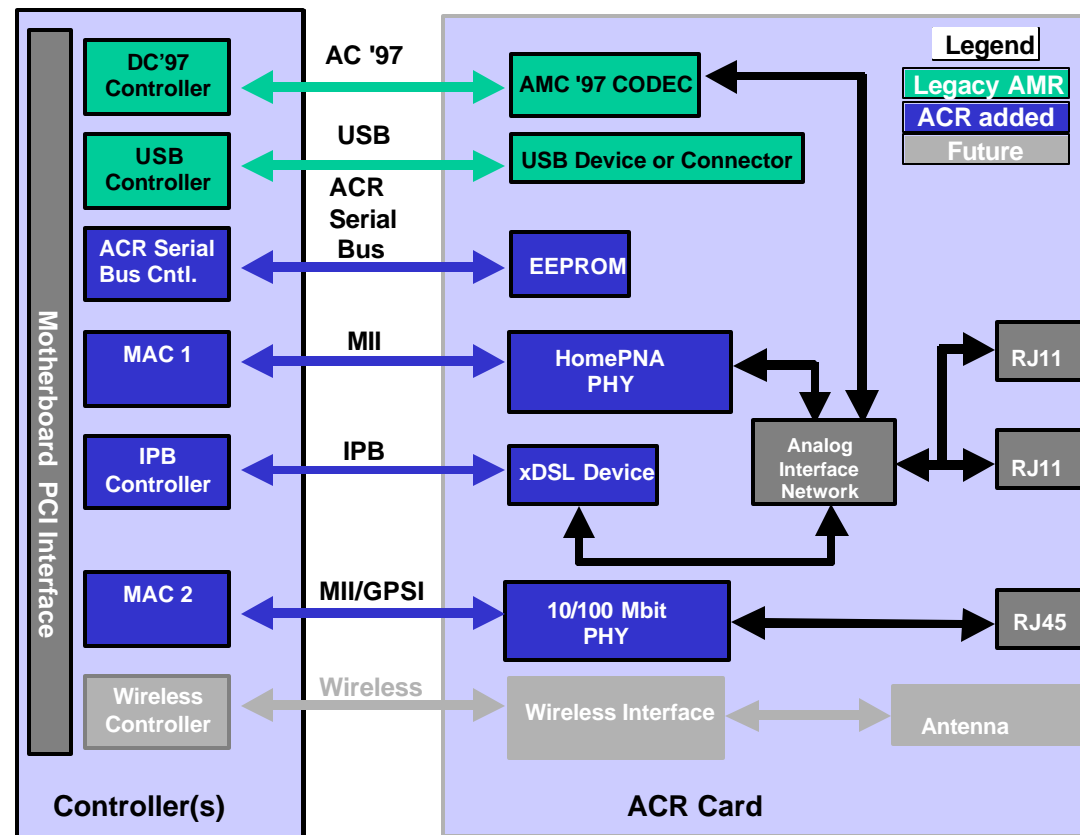
Inventory

- **Motherboard resources that are riser related are listed within the BIOS storage:**
 - Chipset controllers (network, audio, modem, IPB, etc.)
 - Motherboard down controllers (network, etc.)
 - Normally, each appears as a PCI based controller
- **ACR risers should be designed to contain a serial ROM describing the devices on the riser in a flexible format:**
 - Network phys, modem codecs, audio codecs
 - USB devices, serial devices, IPB devices
 - Each device is attached to one of the defined ACR busses

ACR Device Inventory

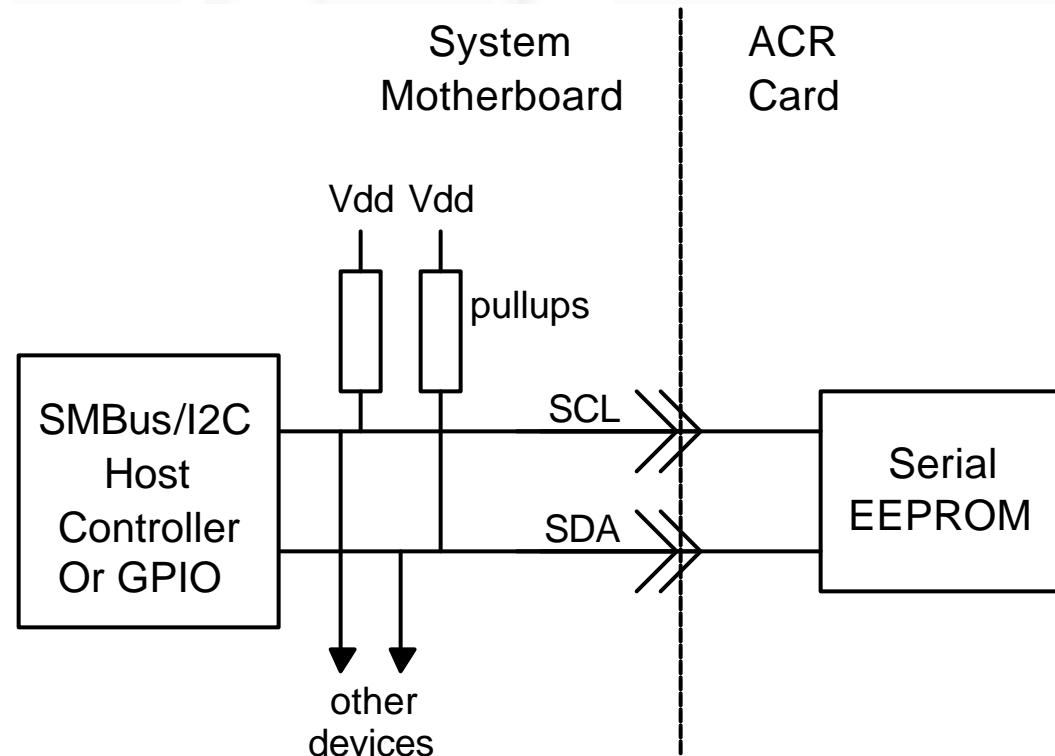
- Devices are listed by bus and function

- Bus: AC-Link, USB, ACR Serial, Primary MII, Secondary MII, GPSI, 7WS, IPB
- Function: Riser, Sound, Modem, HPAN, Ethernet, DSL, Wireless, Cable Modem, Other



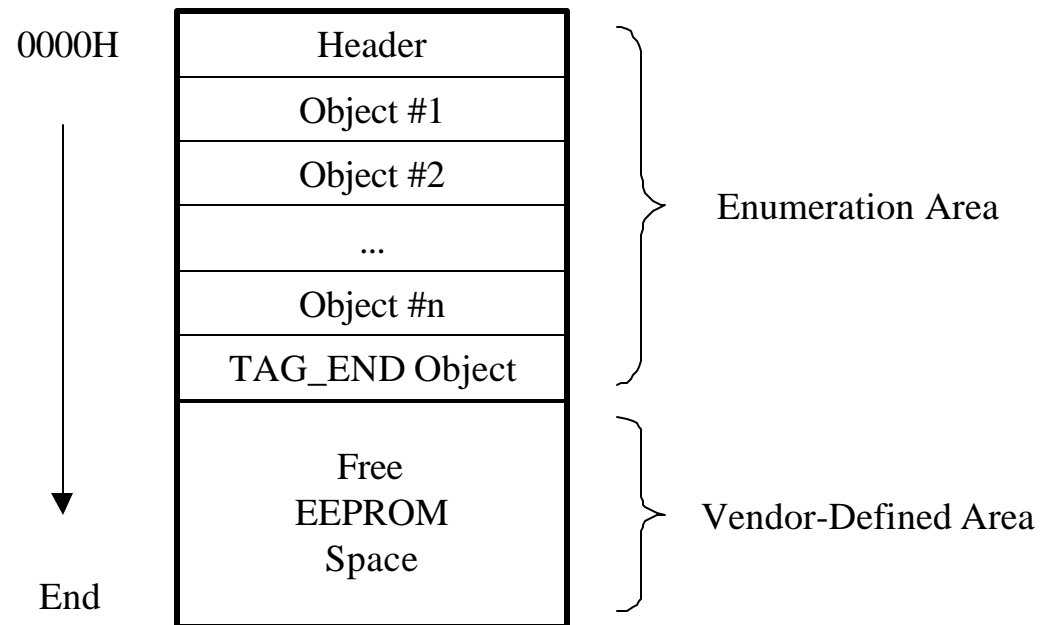
ACR Serial Bus

- Two wire SMBus 1.0 compliant
- Serial Data and Clock, no third control signal
- Eprom at A7 or B0
- Does not require a controller – can also easily use GPIO



Serial EPROM Data

- EEPROM data image looks similar to a linked list
- Data format is variable length



“Marriage”

- **Each riser resource is “married” to a corresponding motherboard resource**
 - Mapping determined by ACR “function number” – one function per bus
 - Additional requirements may be placed on PCI vendor ID in EPROM
 - Additional requirements may be contained by executable code in EPROM
 - ACR runtime environment provided by BIOS
- **Complete devices are composed of the motherboard and riser halves (and will be configured with unique and proper identities)**

Configuration

- **Configuration of motherboard hardware to match riser**
 - Vendor and device ID copied from riser to PCI config space.
 - Subsystem vendor and ID copied from riser to PCI config space.
 - Optionally, PCI class, PCI latency, and PCI power management can also be configured.
 - Controllers disabled if no riser resource is present.
 - Controllers that may not be functional are disabled (e.g., AC-97 configuration issues such as a primary/secondary clock source conflicts).

Net Result of Enumeration

- All special activity is pre-boot, making it compatible with existing OS
- Backward compatible with AMR
- BIOS not specific to riser
- Unique plug and play id (all four numbers) are enabled, a combination of motherboard/riser id
- “Marriages” of risers to controllers can be restricted by either “party”
- Unused components in highly integrated chipset/motherboard pose fewer issues
 - Invisible to OS and user

Examples

- **Useful enumeration pairings:**
 - Modem codec works with all generic controllers
 - MAC on motherboard works only if a compatible PHY on riser
 - HPNA PHY works if only of several compatible controllers on motherboards
 - DSL controller on motherboard works only if specific riser in system
- **If it doesn't work, it doesn't appear to OS**

ACR Enumeration

- Introduction
- **Enumeration**
 - Architectural role
 - **Value**
 - Procedure
 - Examples
 - Going further
- Call to action

Value From User POV

- **Highly integrated peripherals are easier to use and more predictable (fewer configuration hassles and resource conflicts)**
- **Fewer confusing “dead” connectors or devices visible on the OS**
- **Fewer confusing BIOS enable/disable settings, depending on version of product purchased**
- **Single connector when appropriate (shared V.90 and HPNA connector)**
- **Use of riser can free multiple PCI bus slots for use by other peripherals**

Value From IHV POV

- Very simple method for chipset/motherboard component vendors to make hardware generic to many “PHY” vendors and/or “IP” vendors
- Very simple for PHY/riser component vendors to control the combinations of controllers with which the component is paired
- Very simple for PHY/riser component vendors to make sure their hardware IP is protected
- Unique plug and play ids are enabled

Value From ISV POV

- **Helps make sure the right driver is loaded based on riser components**
- **Helps make sure the driver is only loaded when appropriate riser is present**
- **Designed to blaze a path to “unbundle” driver/riser WHQL submissions from motherboard**
- **Designed to “unbundle” driver development from one another (multi-function can be avoided)**
- **Designed to eliminate necessity for driver to probe the codec/PHY components on the fly using indirect registers that are often chip specific**

Value From OEM POV

- **“Sell-up” without functional interaction**
- **Faster WHQL submissions since riser and motherboard may be independently tested**
- **Quicker SKU refresh since risers can be much faster to design/build than motherboards**
- **Leverage high degree of integration without having to put every component and every connector on the motherboard**
- **Free PCI slots to enable other sell-up cards, insurance “space” and/or reduce form factors**

One Motherboard – Scalable Riser



Desktop

- V.90
- 2 Ch audio



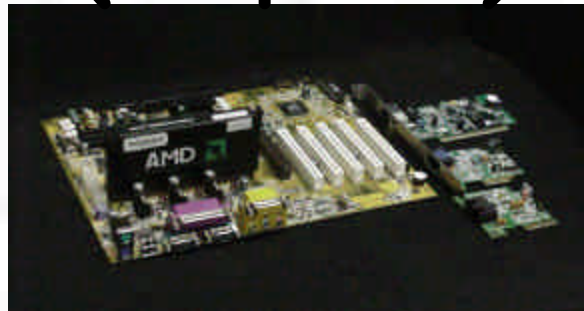
Residential Client

- V.90
- 10/100Base-T
- HomePNA 1 or 2
- 2 Ch audio



Residential Gateway

- V.90
- 10/100 base-T
- HomePNA 1 or 2
- DSL/Cable
- Multi-Channel Audio



ACR Motherboard



ACR Cards

- ACR.Basic - Desktop
- ACR.Lite - Client
- ACR.Hub - Gateway

Value From Channel POV

- **“Sell-up” via full function risers**
- **Last minute configuration and build to order can be implemented without BIOS settings, dependencies, etc**
- **Connectors can be added on the riser rather than on the motherboards while at the same time leveraging high integration chipsets**

Sell-up



- **ACR.Basic**



- **ACR.Lite**



- **ACR.Hub**



One motherboard supports multiple SKUs

ACR Enumeration

- Introduction
- **Enumeration**
 - Architectural role
 - Value
 - **Procedure**
 - Examples
 - Going further
- Call to action

Procedure

- **Basic sequence of enumeration events**
 - Hardware steps
 - BIOS steps
 - OS steps
 - Driver steps

Hardware & BIOS

- **Hardware:**
 - At power up/reset all PCI config registers are set to their defaults
 - Enable SMBus or GPIO
- **Under BIOS control:**
 - Search for ACR EEPROM
 - Validate and store EPROM contents
 - Assign the parameters specified in the EEPROM, plus its own defaults to the PCI config space of each controller supported
 - Disable all ACR controllers that do not have devices attached to them
 - Make all applicable config space registers read-only

OS & Driver

- **Under operating system control:**
 - Scan the PCI bus and matches .INF files to the ID registers in the configuration space
 - Load the appropriate device driver, pointed to by its entry in the INF file
- **Under control of the driver:**
 - Query the BIOS for ACR BIOS presence
 - Query the BIOS for the default IDs of ACR function controllers
 - Query the BIOS for the enumeration data or read it from the serial bus

Goals from Riser POV

- Riser should be designed to enumerate in any motherboard
- EPROM data causes controller to be disabled unless motherboard contains an appropriate controller
- Riser can be tested with ACR specified WHQL Riser Assertions Tests and ACR designated reference motherboards

Goals from Motherboard POV

- Any riser (AMR or ACR) can be plugged and and it should be designed to enumerate fully
- Motherboard can be tested with ACR specified WHQL Motherboard Assertions Tests and ACR designated reference risers

ACR Enumeration

- Introduction
- **Enumeration**
 - Architectural role
 - Value
 - Procedure
 - **Examples**
 - Going further
- Call to action

AMR Compatible Modem

Bytes	EEPROM Contents
4	“ACR 1 0”
2	Riser Object
5	Device ID of riser maker (PCI ID)
19	Nice name of riser: “ACME Turbo Modem”
2	Modem on AC-Link Object
5	Desired Subsystem Ids.
3	END object (with checksum)

- **Shows AMR style, transferring only subsystem IDs**
- **May also include other optional items:**
 - **Riser: serial number**
 - **ACR Serial bus: size of eeprom**
 - **AC-Link Modem: device checks if riser not compatible with all controllers**

Sound & Modem

Bytes	EEPROM Contents
4	“ACR 1 0”
2	Riser Object
5	Device ID of riser maker (PCI ID)
19	Nice name of riser: “WHIZ Audio Modem”
2	Audio on AC-Link Object
5	Desired Subsystem Ids (2).
2	Modem on AC-Link Object
5	Desired Device Ids (2)
3	END object (with checksum)

- **May include optional items like previous example**
- **If audio features are proprietary, may include device checks to make sure right controller is present**

Modem & HPNA

Bytes	EEPROM Contents
4	“ACR 1 0”
2	Riser Object
5	Device ID of riser maker (PCI ID)
24	Nice name of riser: “WHAMMO Modem HPNA G92”
2	Modem on AC-Link Object
9	Desired Device IDs (2).
2	Home Networking on Primary MII Object
5	Desired Subsystem Ids (2)
26	Device Check against three compatible controllers
5	Mac Address
3	END object (with checksum)

- **Shows use of device check**
- **May also include optional latency, class, and power management values of PCI controller**

Modem, HPNA & Ethernet

Bytes	EEPROM Contents
4	“ACR 1 0”
2	Riser Object
5	Device ID of riser maker (PCI ID)
21	Nice name of riser: “BANGO Home Gateway”
2	Modem on AC-Link Object
5	Desired Device IDs (2).
2	Ethernet on Primary MII
9	Desired Device Ids (4)
5	MAC Address
2	Home Networking on Secondary MII
9	Desired Device Ids (4)
5	MAC Address
3	END object (with checksum)

- **Note the easy modularity!**

ACR Enumeration

- Introduction
- **Enumeration**
 - Architectural role
 - Value
 - Procedure
 - Examples
 - **Going further**
- Call to action

Protecting IP

- ACR is about fostering innovation
- Enumeration can help protect your IP
- Protection available:
 - Software IP
 - Device hardware IP
 - Controller hardware IP
- Protection can also be used to help limit support costs to only those situations intended
 - Note that providing write mode for riser EEPROM is not required (may also be a ROM or emulation device)

Ideas for Protecting Software IP

- Protecting software means device driver must function only with a predefined set of hardware
- If using a proprietary controller,
 - Hardwire the controller and base the driver on either 2 IDs (for a driver family) or all 4 PCI IDs
- If using generic controllers (or multiple controllers),
 - Implement “device check” to disable the controller unless it contains one of the vendor/device IDs specified in the riser EPROM
- Driver may also check riser EPROM through BIOS call

Ideas for Protecting Devices

- **Protecting device hardware means making sure it can only function with a specific driver or controller**
- **To make certain a particular driver is loaded,**
 - **Specify that BIOS configure and check the proper device/vendor ids of the PCI controller**
- **To make certain that a particular controller is used,**
 - **Implement “device check” to disable the controller unless it contains one of the vendor/device ids specified in the riser EPROM**

Protecting Controllers

- Protecting a controller means making the controller visible only if particular risers are present or if a specified driver is loaded
- To ensure that only one driver is loaded,
 - Hardwire the device id and specify the driver from it in the inf file
- To ensure that only a particular riser is present,
 - BIOS can contain a very specific list of acceptable risers, based on the vendor id of the riser itself

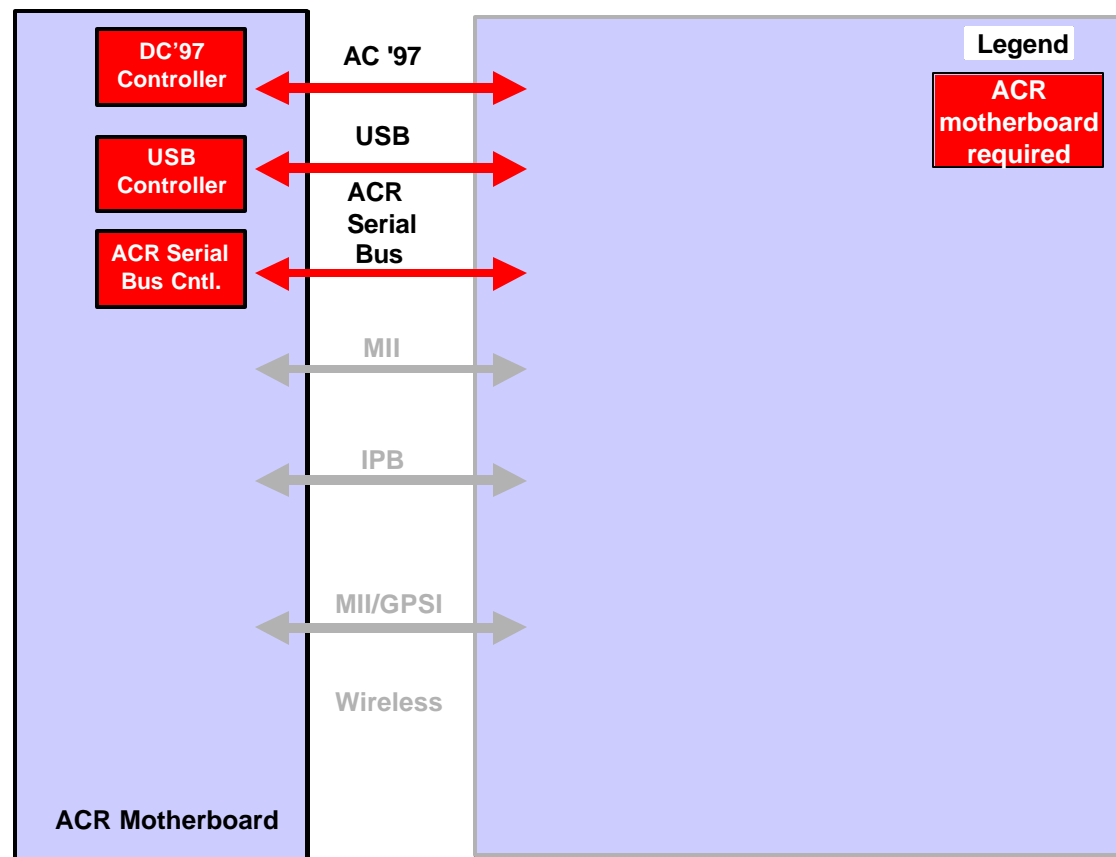
ACR Enumeration

- Introduction
- Enumeration
 - Architectural role
 - Value
 - Procedure
 - Examples
 - Going further
- **Call to action**

Motherboard Call to Action

- **Use ACR instead of AMR**
- **Provide the minimum connections: USB, ACR serial, and AC-97**
- **Always ship motherboard with ACR BIOS!**
- **Don't use designs that allow insertion of devices that conflict with ACR EPROM address**
- **Move SKU dependent connectors to the riser**
- **Move all basic level hardware that is “disabled” in a “sell-up” situation to the riser**
- **Make use of protection modes as appropriate**

Minimum ACR Motherboard



Motherboard Call to Action 2

- Follow ACR 1.0 spec, ACR app notes, AC-97, MII, and IPB
- Participate in the next plugfest, Feb 12 in Taipei
- Submit motherboards to WHQL (AMR style logos) with one of the ACR SIG tested risers
 - They may be considered as candidates for reference boards to test risers independently
- Pre-test risers against:
 - ACR designated risers, ACR specified WHQL motherboard assertions document, WHQL system tests & each appropriate WHQL functional test (modem, audio, networking, etc.)

Riser Calls to Action

- **Use ACR instead of AMR**
- **Provide ACR Serial bus plus busses for your functions**
- **Always ship risers with fully programmed EPROMs, including the riser company vendor id**
- **Always strive to generate unique and complete plug and play ids, relying on enumeration where appropriate**
- **Always locate the serial EPROM at A7 or B0 address**

Riser Calls to Action 2

- Use the ACR EPROM programming tool to generate error-free EPROM images
- Don't put Ethernet PHY or audio codec down with connectors on riser (put the PHY/codec on riser with connector)
- Design all resources so their idle power fits within one PCI idle slot power budget, in case they are unpaired

Riser Call to Action 3

- Follow ACR 1.0 Spec, ACR App Notes, AC-97, MII, and IPB
- Participate in the next plugfest, Feb 12 in Taipei
- Submit risers to WHQL (AMR style logos) with one of the ACR SIG tested motherboards
 - They can be candidates for reference risers to test motherboards independently
- Pre-test risers against ACR designated motherboards, ACR specified WHQL Riser Assertions document, & each appropriate WHQL functional test (modem, audio, networking, etc.)

BIOS Guidelines

- Provide AMR style enumeration as a fallback when AMR card inserted into ACR slot
- Provide full description of any motherboard components that share the busses that connect to ACR slot
- Provide all ACR BIOS functions in both protected mode and 32-bit format

BIOS Guidelines 2

- Follow the ACR BIOS guidelines app note
- Follow ACR 1.0 spec, ACR app notes, AC-97, MII, and IPB
- Participate in the next plugfest, Feb 12 in Taipei
- Pre-test BIOS against ACR specified WHQL motherboard assertions document

Driver Guidelines

- Use separate vendor/sub-vendor ids when appropriate to assure the right driver is loaded
- Provide full and appropriate power management of riser resources through PCI power management
- Make use of the ACR BIOS function to query standard and private information (protected and 32-bit)
- Provide full support for as many combinations of controllers and risers as appropriate

Driver Guidelines

- **Make use of the serial EPROM features to restrict that risers are only used with appropriate controllers**
- **Make use of the serial EPROM features to restrict that drivers are only used with appropriate risers**

Additional Information

- www.acrsig.org to view public information
- info@acrsig.org to join and gain access to members only access and plugfest